CSCI 7000 - Project Report

Student Name: Nitin Kumar
Project Instructor: Claire Monteleoni

Title

Design and evaluation of improvements for modeling high-dimensional probabilistic multivariate time series by improving the underlying flow model.

Introduction

Time series forecasting is a key solution in solving problems related to business as well as engineering. Modeling multivariate time series model poses the problem of scaling to higher dimensions due to the base parametric distribution not being able to model complex data distributions especially in higher dimensions.

Kashif Rasul et al., 2021 proposes "end-to-end trainable autoregressive deep learning architectures for *probabilistic forecasting* that explicitly models multivariate time series and their temporal dynamics by employing a normalizing flow, like the Masked Autoregressive Flow (Papamakarios et al., 2017) or Real NVP (Dinh et al., 2017). These models can scale to thousands of interacting time series."

Underlying flow model is the key to model the statistical dependencies of the multivariate time series data and its improvement, in effect, can lead to modeling of more complex data better. Flow++ (Ho et al., 2019) is a new flow-based model for unconditional density estimation. " It ensures to close the gaps brought by three modeling inefficiencies in prior work on flow models. (1) uniform noise is a suboptimal dequantization choice that hurts both training loss and generalization; (2) commonly used affine coupling flows are not expressive enough; (3) convolutional layers in the conditioning networks of coupling layers are not powerful enough."

Real NVP flow and MAF are the underlying flow models choices in which Real NVP lacks in performance over MAF due to lack in density modeling performance. Recent improvements to flows, e.g. Flow++ (Ho et al., 2019), tries to reduce the performance gap between flow models and autoregressive models and thus provides improved performance compared to Real NVP.

The design choices of the flow model has the capability to improve the performance for modelling of high-dimensional probabilistic multivariate time series. Thus, Flow++ is a strong choice which can improve the modeling of multivariate temporal dynamics.

The Underlying flow model is the key to model the multivariate probabilistic density for the given time series data and below are the current methods which have outperformed RealNVP in all tasks and a strong contender for MAF as well.

Related Work

Neural Spline Flow (Durkan et al, 2019) proposes "a fully-differentiable module based on monotonic rational-quadratic splines, which enhances the flexibility of both coupling and autoregressive transforms while retaining analytic invertibility. It demonstrats that neural spline flows improve density estimation, variational inference, and generative modeling of images". Although, It still stays behind Flow++ in the generative modeling.

Glow (Diederik P. Kingma et al, 2018) as stated in the given paper "is a simple type of generative flow using an invertible 1 × 1 convolution. It demonstrates a significant improvement in log-likelihood on standard benchmarks. Perhaps most strikingly, it demonstrates that a generative model optimized towards the plain log-likelihood objective is capable of efficient realistic looking synthesis and manipulation of large images." It too remains quite far behind in the generative modeling of images in comparison to Flow++.

Flow++ (Ho et al., 2019) provides the following design choices which provides improvement over the other current normalizing flow methods:

(1) " Variational flow-based dequantization instead of uniform dequantization " (Ho et al., 2019)

(**2) "Logistic mixture CDF coupling flows "** (Ho et al., 2019)

(3) "Self-attention in the conditioning networks of coupling layers" (Ho et al., 2019)**.**

Flow ++ Ablation Study (Ho et al., 2019)**:**

The Study proves that uniform dequantization improves the model the most by a large margin (Ho et al., 2019).

Ablation	bits/dim	parameters
uniform dequantization	3.292	32.3M
affine coupling	3.200	32.0M
no self-attention	3.193	31.4M
Flow++ (not converged for ablation)	**3.165**	**31.4M**

Source: (Ho et al., 2019) , Table 2

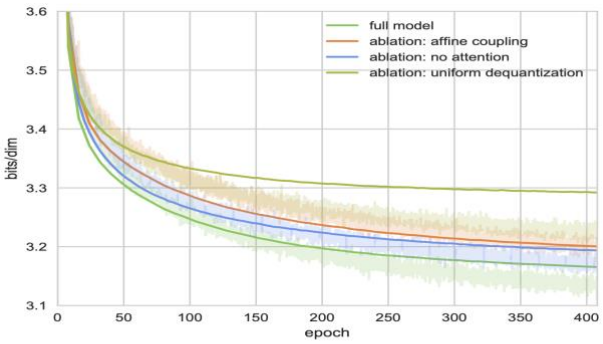

Figure 1. Ablation training (light) and validation (dark) curves on unconditional CIFAR10 density estimation. These runs are not fully converged, but the gap in performance is already visible.

Source: (Ho et al., 2019) , Figure 1

Methodology/Results

Evaluation Method: Continuous Ranked Probability Score (CRPS) is used to evaluate the generated time series models.

The Project started with the study of current literatures ie. Flow++ (Ho et al., 2019), Glow (Diederik P. Kingma et al, 2018), Neural Spline Flow (Durkan et al, 2019) . Flow++ showed a very strong promise of improvements due to its design choices.

I first validated the results of Temporal Conditioned normalizing flows implemented here for the case of both RealNVP and MAF using the metrics CRPS. It showed the comparable results as mentioned in the paper and documented in its official package.

Next, I went ahead and implemented the official version of Flow++ in the Temporal conditioned normalizing flow by the replacement of flow used (RealNVP/MAF). Official version of Flow++ is implemented in Tensorflow 1.10.x which needed back support of API to be integrated to the main Temporal conditioned normalizing flow package. Also, after its implementation, it still was incompatible during the training phase due to Temporal conditioned normalizing flow package being implemented in PyTorch and it being in Tensorflow.

I moved to an unofficial version of flow++ called flowplusplus. It is implemented in PyTorch and was suitable to be used, but it needed validation, so trained and tested it for CIFAR dataset and it gave the required results.

Then, I moved on to use flowplusplus package and implemented it to replace the flow in Temporal conditioned normalizing flow. It was compatible. Then, I worked on the data layer of the model since the flow++ is used for image dataset. I updated the code to use map the dimensions of time series formatted data to be used in the given model. Although, Flow++ uses convolution layers especially in multi-layer architecture due to which data mapping is becoming incompatible even at the internal convolution layers which needs further development to be updated and tested.

Validation of different flow++ packages and its implementation were done for Temporal Conditioned normalizing flows. Since, Flow++ uses convolution layers also in multi system architecture fashion, it would require changes in the internal convolution layers to be used for the time series modeling.

Conclusion

At the current state, with the validation of the different flow++ packages, it is evident that it is strong contender to improve the density modeling. The Ablation study shows the strong results especially due to dequantization which other flow models do not consider. Other design choices also have improvements as could be observed from the results mentioned above.

Further, as a next step, Convolution layer architecture in the flow++ needs to be tuned to solve the use case of multi variate time series modeling.

Also, rather than multi system architecture, a simplified version of it stacked might be more helpful here for the case of time series modeling since multi system architecture uses the layers of splitting and joining (using checkerboard and other methods) which is good for generalization in case of images but for time series data the merging of data may rather lose the information that it represented which is not the case for the image data.

Also, An Ablation study needs to be done to see the improvements made by dequantization as well as other design choices included in flow++.

As a next level progress, when the first part of flow is improved, the attention layer can also be improved using the current state of the art methods to improve the conditioning thus to better model the multi variate time series data.

Bibliography

Citations:

1. Ho, J., Chen, X., Srinivas, A., Duan, Y., & Abbeel, P. (2019). *Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design*. URL: https://arxiv.org/abs/1902.00275
2. Kingma, D. P., & Dhariwal, P. (2018). *Glow: Generative Flow with Invertible 1x1 Convolutions*. URL: https://arxiv.org/abs/1807.03039
3. Durkan, C., Bekasov, A., Murray, I., & Papamakarios, G. (2019). *Neural Spline Flows*. URL: https://arxiv.org/pdf/1906.04032.pdf

4. Rasul, K., Sheikh, A.-S., Schuster, I., Bergmann, U., & Vollgraf, R. (2021). *Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows*. URL: https://arxiv.org/pdf/2002.06103

Data Links :

1. Taxi trajectory prediction, URL: https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i
2. Electricity load diagrams, URL: https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
3. PEMS-sf dataset, URL: https://archive.ics.uci.edu/ml/datasets/PEMS-SF
4. Compiled time series data sets, URL: https://github.com/mbohlkeschneider/gluon-ts/tree/mv_release/datasets

Relevant Code/Resource Links:

5. Deep Learning Time Series Forecasting, URL: https://github.com/Alro10/deep-learning-time-series
6. Flow-based Deep Generative Models, URL: https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html
7. PyTorchTS, URL: https://github.com/zalandoresearch/pytorch-ts
8. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, URL: https://github.com/aravindsrinivas/flowpp
9. Flow++ in PyTorch, URL: https://github.com/chrischute/flowplusplus
10. Flow++ implementation in PyTorchTS, URL: https://github.com/nitinthedreamer/pytorch-ts